



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY
FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

CLUSTERING MRAKU BODŮ

POINT CLOUD CLUSTERING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DANIEL MRKVIČKA

VEDOUCÍ PRÁCE
SUPERVISOR

Mgr. JANA PROCHÁZKOVÁ, Ph.D.

BRNO 2017

Zadání bakalářské práce

Ústav: Ústav matematiky
Student: **Daniel Mrkvička**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Matematické inženýrství
Vedoucí práce: **Mgr. Jana Procházková, Ph.D.**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Clustering mraku bodů

Stručná charakteristika problematiky úkolu:

Student se seznámí s různými metodami zpracování mraku bodů, provede jejich srovnání a vypíše výhody a nevýhody. Na zkušebních datech (reálná data z 3D scanneru) poté zpracuje některý z algoritmů a provede testování pro mezní případy. Také provede testování různých existujících programů, aby srovnal jejich funkčnost a stabilitu.

Cíle bakalářské práce:

1. Teoretické shrnutí různých metod třídění bodů - z literatury.
2. Srovnání jednotlivých metod.
3. Programové zpracování jedné (více) metod a její testování.

Seznam literatury:

FISCHLER, Martin A. a Robert C. BOLLES. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM [online]. 24(6), 381-395 [cit. 2016-11-09]. DOI: 10.1145/358669.358692. ISSN 00010782. Dostupné z: <http://portal.acm.org/citation.cfm?doid=358669.358692>

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17

V Brně, dne

L. S.

prof. RNDr. Josef Šlapal, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato bakalářská práce se zabývá zpracováním mraku bodů. Zaměřuje se na detekci ploch v trojrozměrném prostoru. Popisuje především metody, které jsou používány k detekci rovin. Dále popisuje konkrétní implementaci jedné z těchto metod, metodu RANSAC, a zkoumá její praktické použití k detekci střech.

Abstract

This bachelor's thesis deals with the point cloud clustering. It focuses on surface detection in three-dimensional space. It describes in particular the methods that are used for plane detection. It also describes the concrete implementation of one of these method, the RANSAC, and examines its practical application for roof detection.

klíčová slova

Clustering mraku bodů, RANSAC, Houghova transformace, paralelní souřadnice, detekce roviny

keywords

Point cloud clustering, RANSAC, Hough transform, parallel coordinates, plane detection

Prohlašuji, že jsem bakalářskou práci *Clustering mraku bodů* vypracoval samostatně pod vedením Mgr. Jany Procházkové, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Daniel Mrkvička

Chtěl bych poděkovat své vedoucí práce Mgr. Janě Procházkové, Ph.D. za odborné vedení během tvorby bakalářské práce. Dále bych chtěl poděkovat svým rodičům za podporu během celého mého dosavadního studia.

Daniel Mrkvička

Obsah

1	Úvod	12
2	Matematický aparát	13
2.1	Vlastnosti roviny	13
2.2	Práce s nalezenými rovinami	14
2.2.1	Rotace bodů v prostoru	14
2.2.2	Andrew's monotone chain	15
2.2.3	Gaussova metoda pro výpočet obsahu n-úhelníku	15
3	Metody	16
3.1	Metoda nejmenších čtverců	16
3.1.1	Proložení bodů rovinou	16
3.2	RANSAC	17
3.2.1	Popis algoritmu	17
3.2.2	Použití	18
3.2.3	Metody vycházející z algoritmu RANSAC	19
3.3	Houghova transformace	20
3.3.1	Transformace bodu pro detekci rovin	20
3.4	Paralelní souřadnice	22
3.4.1	Vlastnosti	22
3.4.2	Využití pro clustering	23
3.4.3	Paralelní prostor	24
3.5	Srovnání metod	24
4	Implementace RANSACu	26
4.1	Cíle implementace	26
4.2	Způsob implementace	26
4.3	Rozdíl mezi implementací a původním algoritmem	26
4.4	Vstupy	27
4.5	Výstupy	27
4.6	Popis algoritmu	27
4.7	Grafické rozhraní a práce s programem	27
4.8	Testování na reálných datech	28
5	Solární kalkulačka	32
5.1	Teorie	32
5.2	Panel solární kalkulačky	37
5.3	Způsob vyhodnocení získaných dat	37
6	Závěr	40
7	Seznam příloh	43

1 Úvod

Neustále se snažíme rozvíjet způsob, jakým jsme schopni měřit a zaznamenávat svět kolem nás. Velký rozvoj tak zaznamenaly i 3D senzory a 3D skenery (zařízení, která snímají povrch jako množinu bodů - výsledkem je tzv. mrak bodů). Dá se říci, že pomocí nich jsme už schopni zaznamenat okolní prostředí s velmi vysokou přesností. Zároveň se tyto přístroje stávají levnější a dostupnější, zmiňme například pohybový senzor Kinect, který je součástí téměř každé herní konzole X-box One.

Lze tedy předpokládat, že v budoucnu budou některé stroje (například autonomní auta či tovární roboty) často vnímat své okolí ve třech dimenzích. Proto je velice důležité, aby se rozvíjely i metody, které nám umožní naplno využívat možnosti 3D snímání. Požadavkem na tyto metody je schopnost pracovat velice rychle s velkým množstvím dat a dosahovat co nejmenšího množství chyb.

Těmito metodami se zabývá obor počítačového vidění. Jedná se o obor výpočetní techniky, který se zabývá snímáním okolního prostředí, získáváním informací v podobě strojového kódu a jejich následným zpracováním. Snímat okolí je možné například pomocí klasické kamery nebo 3D senzorů. Počítačové vidění se tak zabývá analýzou jak 2D obrazu, tak 3D scény.

Jedním ze způsobů práce s mračenem bodů je clustering, který slouží k třídění bodů podle vybrané vlastnosti těchto bodů. Právě clusteringem se bude zabývat bakalářská práce a za společnou vlastnost bodů se bude uvažovat to, že body leží ve stejném geometricky popsatelném útvaru.

Bakalářská práce má za cíl přiblížit geometrii roviny a popsat několik metod, které umožňují její detekci v mraku bodů.

Práce je rozdělena do šesti kapitol. Kapitola 2 popisuje základní vlastnosti roviny a dále některé vybrané metody sloužící k následné úpravě rovin po jejich nalezení.

Kapitola 3 obsahuje popis čtyř vybraných metod, které lze použít k vyhledání rovin v mraku bodů. Každé metodě je věnována jedna podkapitola. První metodou je metoda nejmenších čtverců, která není používána samostatně, ale je často součástí ostatních popsaných metod. Další metodou je často využívaná metoda RANSAC. V příslušné podkapitole jsou také uvedeny některé z mnoha modifikací, které se používají pro dosažení lepších výsledků. Další podkapitoly jsou věnovány Houghově transformaci a metodě paralelních souřadnic. Metoda paralelních souřadnic je uvedena spíše jako zajímavost a na svoje větší využití v rámci clusteringu zatím čeká.

Kapitola 4 se zabývá popisem programu, který využívá RANSACu k detekci střech na domech. Je zde popsáno prostředí programu, základní vnitřní logika a výsledky testování. Testována byla schopnost programu zpracovat vlastnoručně naměřená data i data z leteckého skenování. Dále je zde popsán způsob práce se skenerem ATOS, který byl použit k naměření vlastních dat. V závěru kapitoly jsou dosažené výsledky porovnány s profesionálním softwarem.

V kapitole 5 jsou popsány fyzikální veličiny, které jsou použity k výpočtu množství dopadajícího slunečního záření na plochu rovné střechy. Popsané závislosti jsou implementovány v programu a nabízí rychlé vykreslení grafů denních hodnot. Kapitola proto popisuje i vyhodnocení získaných dat.

Kapitola 6 shrnuje celou práci a hodnotí dosažené výsledky.

2 Matematický aparát

Pro mrak získaný během skenování skutečného objektu pomocí standardního 3D skeneru platí, že získanými informacemi o jednotlivých bodech jsou nejčastěji souřadnice bodu, normála povrchu v nejbližším okolí bodu, případně barva tohoto okolí.

My se zaměříme pouze na souřadnice bodu a na práci s nimi. Každý nasnímaný bod má obvykle tři souřadnice, budeme proto v celé práci, nebude-li uvedeno jinak, uvažovat prostor \mathbb{R}^3 .

Dále se v celé práci omezíme na identifikaci rovin. Jedná se o plochy, které jsou jednoznačně určeny již třemi body. Většina metod, kterými se budeme zabývat, vychází z řešení problému nalezení přímek v prostoru \mathbb{R}^2 .

2.1 Vlastnosti roviny

Metody clusteringu využívají řadu poznatků z analytické geometrie a lineární algebry. Zde shrneme pouze některé významnější vlastnosti rovin. Více lze nalézt například ve skriptech lineární algebry [1], stejně jako důkazy vět v této kapitole.

Definice 2.1. *Normálovým vektorem roviny v prostoru se rozumí libovolný vektorový nenulový vektor \mathbf{n} , jehož umístění jsou kolmá k dané rovině.*

Věta 2.2. *Množina všech bodů $X = [x, y, z]$ v prostoru, pro něž platí*

$$ax + by + cz + d = 0,$$

kde alespoň jedno z čísel a, b, c není nula, je rovina s normálovým vektorem $\mathbf{n} = (a, b, c)$.

Předchozí rovnice se nazývá *obecná rovnice roviny*

Věta 2.3. *Bod $X = [x, y, z]$ leží v rovině ρ procházející bodem $A = [a_1, a_2, a_3]$ a rovnoběžné s lineárně nezávislými vektory $\mathbf{s}_1 = (s_{11}, s_{12}, s_{13})$, $\mathbf{s}_2 = (s_{21}, s_{22}, s_{23})$, právě když existují čísla $u, v \in \mathbb{R}$ taková, že platí:*

$$\begin{aligned}x &= a_1 + us_{11} + vs_{21}, \\y &= a_2 + us_{12} + vs_{22}, \\z &= a_3 + us_{13} + vs_{23}\end{aligned}$$

Tyto rovnice se souhrnně nazývají *parametrické rovnice roviny*. Parametrické rovnice lze převést na obecnou rovnici vyloučením parametrů u, v . Z těchto dvou vyjádření roviny budeme používat především obecnou rovnici.

Dále využijeme výpočet vzdálenosti bodu od roviny, neboť právě vzdálenost bývá kritériem toho, jestli daný bod leží v dané rovině či nikoli.

Věta 2.4. *Pro vzdálenost $d(A, \rho)$ bodu $A = [a_1, a_2, a_3]$ od roviny ρ o rovnici $ax + by + cz + d = 0$ platí:*

$$d(A, \rho) = \frac{|aa_1 + ba_2 + ca_3 + d|}{\sqrt{a^2 + b^2 + c^2}}.$$

Při zjišťování, co daná rovina představuje (například střecha nebo stěna domu), je důležitá její orientace vůči jiným vybraným rovinám. Vzájemnou orientaci dvou rovin lze jednoduše vyjádřit pomocí úhlu, který tyto dvě roviny svírají.

Věta 2.5. Pro úhel α dvou rovin ρ_1, ρ_2 s normálovými vektory $\mathbf{n}_1, \mathbf{n}_2$ platí:

$$\cos \alpha = \frac{|\mathbf{n}_1 \cdot \mathbf{n}_2|}{\|\mathbf{n}_1\| \cdot \|\mathbf{n}_2\|}$$

Na závěr této podkapitoly porovnejme obecnou rovnici roviny s obecnou rovnicí přímky v rovině.

Věta 2.6. Množina všech bodů $X = [x, y]$ v rovině, pro něž platí:

$$ax + by + c = 0,$$

kde alespoň jedno z čísel a, b není nula, je přímka s normálovým vektorem (a, b) .

Porovnáme-li tvar obecných rovnic ve větách 2.6 a 2.3, vidíme, že se liší pouze v členu cz , který odpovídá třetí dimenzi. Tato podobnost je jednou z vlastností využitelných pro snadné rozšíření metod, které pracují s přímkou v rovině vyjádřenou v tomto tvaru.

2.2 Práce s nalezenými rovinami

Plochy na skutečných objektech jsou ohraničené. Proto je vhodné tyto plochy popsat jinak než jako nekonečné roviny. Jednoduchý způsob je každou takovou plochu popsat jako n -úhelník s nejmenším možným obsahem ležící v příslušné rovině, který bude obsahovat všechny přiřazené naměřené body.

Budeme-li předpokládat, že takový n -úhelník je konvexní, pak lze množině všech naměřených bodů patřících ke stejné rovině přiřadit konvexní obálku, jejíž vrcholy jsou zároveň vrcholy hledaného n -úhelníku.

Jeden z možných způsobů provedení je nejprve provést stejnou rotaci každého bodu do roviny XY (možné drobné odchylky v ose z zanedbáme). Poté lze použít algoritmus *Andrew's monotone chain* k nalezení konvexní obálky. Nakonec provedeme zpětnou transformaci vrcholů konvexní obálky do původní roviny.

Pro takto určený n -úhelník lze spočítat plošný obsah pomocí Gaussovy metody. Tento výpočet je vhodné provést před zpětnou transformací.

Tyto metody byly použity v implementaci RANSACu (kapitola 4.6).

2.2.1 Rotace bodů v prostoru

Rotační matice popisují jedno ze shodných zobrazení v rovině i prostoru. Vyčerpávající popis nalezneme např. v knize Matematické principy grafických systémů od D. Martiška [3]. Nám postačuje pracovat pouze s rotačními maticemi pro rotaci kolem osy x a z .

Obecně je rotace shodné zobrazení $f : \mathbb{R}^3 \mapsto \mathbb{R}^3$, které každému bodu $X[x, y, z]$ přiřadí bod $X'[x', y', z'] = X \cdot A$, kde A je ortogonální matice. Všechny body prostoru jsou otočeny kolem pevně daného bodu (středu otočení) stejným směrem o stejný úhel (úhel otočení).

Matice pro rotaci kolem osy x o úhel ϕ v kladném smyslu otáčení.

$$A_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix}$$

Matice pro rotaci kolem osy z o úhel ψ v kladném smyslu otáčení.

$$A_z = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

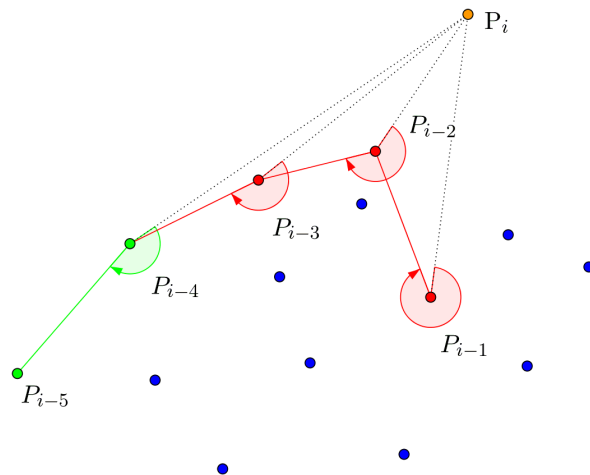
2.2.2 Andrew's monotone chain

Andrew's monotone chain [4] se používá k určení konvexní obálky množiny bodů v \mathbb{R}^2 . Celý proces se dá rozdělit na 3 části: seřazení bodů, složení horní a dolní obálky a sestavení celé obálky.

Nejprve je nutné body uspořádat podle souřadnice x , v případě rovnosti pak podle souřadnice y .

Následně se určí horní obal. Algoritmus začíná v bodě, který má nejmenší x -ovou souřadnici, a postupuje se až do bodu s největší x -ovou souřadnicí. Každý bod P_i se postupně přidá do množiny horního obalu, ale před jeho přidáním se kontroluje, zda tento bod neleží vně dosud vymezené oblasti. Tato kontrola se provádí pomocí výpočtu úhlu, který svírá trojice P_i, P_{i-1}, P_{i-2} (vrcholem úhlu je bod P_{i-1}), kde body P_{i-1}, P_{i-2}, \dots jsou body již přiřazené obálce seřazené podle pořadí přidání (P_{i-1} je poslední přidáný bod). Úhel je počítán ve směru hodinových ručiček a kontroluje se, zda je menší než 180° . Pokud tomu tak není, bude z množiny vrcholů odebrán bod P_{i-1} a výpočet bude opakován pro trojici P_i, P_{i-2}, P_{i-3} , případně pro trojici P_i, P_{i-3}, P_{i-4} atd. Velikost úhlu je kontrolována pouze v případě, že množina horního obalu obsahuje alespoň dva body. Bod mající největší x -ovou souřadnici se do obalu už přidávat nebude (je součástí dolního obalu).

Poté se obdobně sestaví dolní obal. Jako výchozí bod se tentokrát bere bod s největší x -ovou souřadnicí a následně se pokračuje až do bodu s nejmenší x -ovou souřadnicí. Celý konvexní obal je složením horního a dolního obalu.



Obrázek 1: Andrew's monotone chain: před přidáním oranžového bodu P_i dojde k odstranění červených bodů P_{i-1}, P_{i-2} a P_{i-3} z horního obalu

2.2.3 Gaussova metoda pro výpočet obsahu n-úhelníku

Tato metoda [5] slouží k určení obsahu S n -úhelníku s vrcholy $P_1 [x_1, y_1], P_2 [x_2, y_2], \dots, P_n [x_n, y_n]$. V programu se používá k určení obsahu povrchu střech.

$$S = \frac{1}{2} \left| \sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_i y_i - x_1 y_n \right| \quad (2.1)$$

3 Metody

3.1 Metoda nejmenších čtverců

Metoda nejmenších čtverců je jednou ze základních a často používaných metod pro proložení bodů například křivkou nebo plochou. Oproti jiným používaným metodám postrádá schopnost rozeznat chyby nebo více útvarů, a proto se nepoužívá samostatně. Přesto zůstává klíčová pro přesné určení hledaného útvaru po roztrídění bodů, a proto nachází využití v kombinaci s jinými metodami.

3.1.1 Proložení bodů rovinou

Metoda nejmenších čtverců hledá řešení rovnic tak, aby se minimalizoval součet čtverců odchylek vůči těmto rovnicím.

V případě prokládání bodů rovinou budeme vycházet z obecné rovnice roviny ve tvaru

$$A'x + B'y + C'z + D' = 0 \quad A', B', C', D' \in \mathbb{R}. \quad (3.1)$$

Tuto rovnici si upravíme tak, abychom měli co nejméně neznámých koeficientů. Za předpokladu, že koeficient C' je nenulový, ji můžeme upravit do tvaru

$$z = Ax + By + C \quad A, B, C \in \mathbb{R} \quad (3.2)$$

a dostáváme tak pouze tři neznámé koeficienty. Budeme řešit soustavu rovnic (princip odvození lze nalézt např. ve skriptech numerických metod [2])

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}, \quad (3.3)$$

$$\mathbf{A} = \begin{pmatrix} x_1 & y_1 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} A \\ B \\ C \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}.$$

Po dosazení dostáváme soustavu rovnic

$$\begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & \sum_{i=1}^n 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i z_i \\ \sum_{i=1}^n y_i z_i \\ \sum_{i=1}^n z_i \end{pmatrix}, \quad (3.4)$$

která je ve tvaru $\mathbf{A}'\mathbf{x} = \mathbf{b}'$, a její řešení můžeme získat například pomocí Gaussovy eliminační metody.

Před použitím metody je vhodné odhadnout přibližné hodnoty koeficientů v (3.1) pomocí několika bodů. Pokud absolutní hodnota koeficientu C' není větší než absolutní hodnoty koeficientů A' a B' nebo je dokonce rovna nule, je vhodnější hledat rovnici (3.2) ve tvaru $y = Ax + Bz + C$ nebo $x = Ay + Bz + C$ podle toho, který koeficient je větší v absolutní hodnotě. Poté je nutné adekvátně upravit i soustavu lineárních rovnic (3.4). Toto opatření je nutné k zachování co největší přesnosti výpočtu.

3.2 RANSAC

Random Sample Consensus (RANSAC) je metoda, která byla publikovaná roku 1981 dvojicí autorů Martinem A. Fischlerem a Robertem C. Bollesem [6]. Obecně slouží ke zpřesnění modelu, který obsahuje velký počet chyb. Její použití bylo demonstrováno při určení místa pořízení leteckých snímků za pomoci identifikace významných orientačních bodů. V dnešní době se používá zejména v oblasti počítačového vidění a vzniklo mnoho různých variant. Jedná se o nedeterministický algoritmus, neboť zahrnuje náhodný výběr bodů, který může vést k různým výsledkům.

3.2.1 Popis algoritmu

Následující definice je překlad původní definice z článku M. A. Fischlera a R. C. Bollese [6].

Definice 3.1. Nechť je dán model, který potřebuje nejméně n datových bodů k určení jeho neznámých parametrů. Dále nechť je dána množina datových bodů P taková, že $|P| \geq n$. Nechť je náhodně zvolena její podmnožina S_1 mající n bodů a z ní je stanoven model M_1 . Stanovený model nechť je použit k určení podmnožiny S_1^* bodů z P , které leží v nějaké chybové vzdálenosti od M_1 . Množina S_1^* se nazývá *consensus set* množiny S_1 .

Pokud je $|S_1^*|$ větší nebo rovno prahu (threshold) t , kde t je funkce odhadovaného počtu hrubých chyb v P , nechť je S_1^* použita k určení nového modelu M_1^* (například za použití metody nejmenších čtverců).

Pokud je $|S_1^*|$ menší než t , nechť je zvolena podmnožina S_2 a zopakován celý postup. Pokud ani po předem daném počtu pokusů není nalezen *consensus set* s t nebo větším počtem bodů, nechť je za výsledek považován největší *consensus set* nebo je hledání ukončeno neúspěchem.

Každý bod, který náleží množině *consensus set*, se nazývá inlier. Pokud bod v této množině neleží, nazýváme ho outlier.

Algoritmus popsáný v Def. 3.1 lze v některých situacích jednoduše vylepšit. Za prvé, pokud použití náhodného výběru S_i nemá očekávané výsledky, je možné zvolit deterministický postup. Za druhé, na základě modelu M_i^* je možné přidat do množiny S_i^* další body z P a model na základě větší množiny bodů znovu zpřesnit.

Dále je důležité zvolit vhodný počet iterací k . Nechť w značí pravděpodobnost, že se libovolný bod nachází ve vzdálenosti menší než práh t od modelu a n je počet bodů nutných k určení parametrů modelu. Potom lze lehce odvodit, že

$$E(k) = b + 2 \cdot (1 - b) \cdot b + 3 \cdot (1 - b)^2 \cdot b + \dots + m(1 - b)^{m-1} \cdot b + \dots$$

kde $E(k)$ je střední hodnota počtu nutných iterací a $b = w^n$. Položme $a = (1 - b)$

$$E(k) = b \cdot (1 + 2a + 3a^2 + \dots + ma^{m-1} + \dots) \quad (3.5)$$

Pro $a \in \langle 0; 1 \rangle$, platí

$$\sum_{i=0}^{\infty} a^i = 1 + a + a^2 + a^3 + \dots = \frac{a}{1 - a}$$

Po záměně stran a po derivaci výrazu podle a

$$\frac{1}{(1 - a)^2} = 1 + 2a + 3a^2 + \dots \quad (3.6)$$

Při velkém množství bodů dosadíme výraz rovnice 3.6 do rovnice 3.5 a můžeme $E(k)$ aproximovat jako:

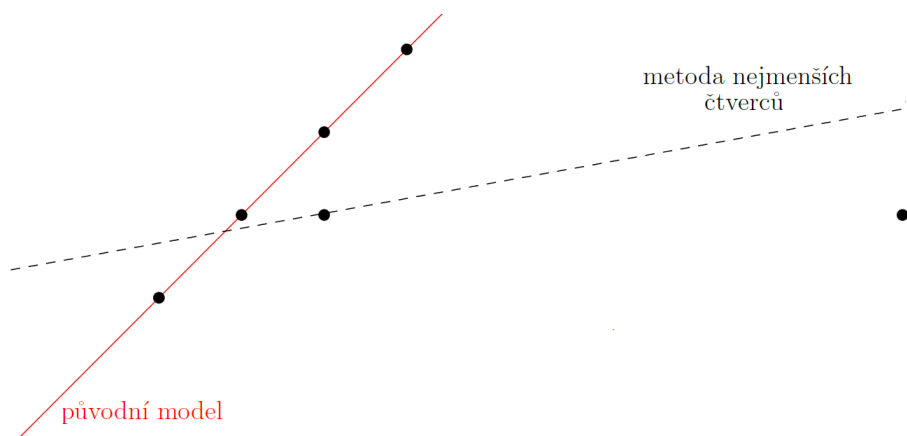
$$E(k) = b \cdot \frac{1}{(1-a)^2} = b \cdot \frac{1}{b^2} = \frac{1}{b} = w^{-n}$$

Obecně je vhodné nastavit počet iterací na hodnotu větší než $E(k)$. Hodnota $E(k)$ závisí na w a tedy na zvoleném prahu.

Práh t musí být dostatečně velký, aby byl nalezen správný model a aby tento model obsahoval dostatečný počet inlierů. Na druhou stranu nesmí být velký příliš, neboť musí zahrnovat minimum chybných bodů. Přesný výpočet pro velikost prahu neexistuje. Při jeho určení je možné například vycházet z přesnosti měřících zařízení použitých k získání množiny bodů.

RANSAC tedy nejdříve zvolí malé množství bodů, které je dostatečné pro určení modelu. Určí předběžný model a poté zjišťuje množství bodů, které leží v dostatečné blízkosti určeného modelu. Pokud nalezne dostatečné množství bodů, zpřesní předběžný model pomocí metody nejmenších čtverců a prohlásí ho za výsledek.

Základní vlastností RANSACu, pro kterou je používán, je schopnost odfiltrout velký počet chybných dat při určování výsledného modelu. Tím se liší například od metody nejmenších čtverců, která k popsání modelu používá všechna dostupná data a může tak dojít ke značnému zkreslení, jak je vidět na obr. 2.



Obrázek 2: Chyba při použití metody nejmenších čtverců

3.2.2 Použití

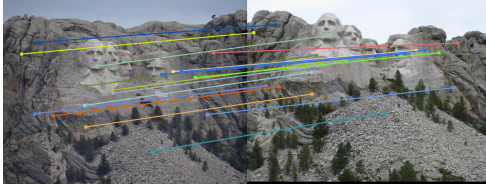
Metoda RANSAC, Houghova transformace i metoda paralelních souřadnic mají velmi podobné oblasti používání. Přesto je metoda RANSAC v současnosti využívána více než zbylé dvě. Proto zde krátce nastíníme některé možnosti jejího použití.

Metoda je hojně používána k detekci rovných čar na fotografiích nebo videu, případně k detekci jiných geometrických obrazců [7].

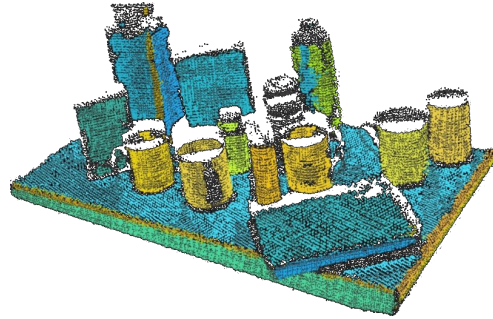
Dále je využívána k porovnání významných orientačních bodů na různých snímcích. Toho se používá například k sestavení panoramat z více fotografií [8]. Metoda také našla využití v epipolární geometrii [9], což je vnitřní projektivní geometrie mezi dvěma pohledy. Vnitřní geometrie je popsána takzvanou fundamentální maticí. Umožňuje sestavit částečné prostorové vidění ze dvou kamer, které snímají stejnou scénu z rozdílných úhlů. RANSAC je zde využit během určování fundamentální matice kvůli odstranění chybně určených dvojic odpovídajících si bodů (obr. 3) [10].

V případě práce s mrakem bodů se RANSAC používá především k rozřídění bodů na jednotlivé objekty (obr. 4) [11]. Tyto objekty se musí skládat z matematicky popsatelných ploch, jako je například rovina a nebo povrchy těles (kulová plocha, válec).

V rámci bakalářské práce bylo RANSACu použito k nalezení rovin.



Obrázek 3: Vybrané odpovídající si body určené RANSACem, převzato z internetu [10]



Obrázek 4: Rozříděný mrak bodů na jednotlivé plochy, převzato z internetu [26]

3.2.3 Metody vycházející z algoritmu RANSAC

Základní algoritmus RANSAC je poměrně jednoduchý. Proto z něj bylo odvozeno mnoho metod, které upravují původní algoritmus různými způsoby. Zde jsou uvedeny jako příklad tři typy úprav, kterými lze výsledky původní metody výrazně vylepšit.

MSAC, MLESAC

RANSAC rozhoduje o správnosti modelu pouze na základě počtu inlierů. Každému nalezenému modelu lze přiřadit jeho správnost následující funkcí, přičemž po správném modelu požadujeme její co nejmenší hodnotu.

$$C = \sum_{i=1} \rho(e_i^2),$$

kde e_i je vzdálenost i -tého bodu od modelu označována také jako chyba, t je práh a ρ je metrická funkce definovaná následovně

$$\rho(e^2) = \begin{cases} 0 & e^2 < t^2 \\ \text{konstanta} & e^2 \geq t^2 \end{cases}$$

MSAC (*m-estimator sample consensus*) používá rozdílnou funkci ρ , která zahrnuje i chybu inlierů [12].

$$\rho(e^2) = \begin{cases} e^2 & e^2 < t^2 \\ t^2 & e^2 \geq t^2 \end{cases}$$

Na MSAC navazuje metoda MLESAC (*maximum likelihood consensus*), která předpokládá, že chyba bodů odpovídá především normálnímu rozdělení [12]. Protože ne vždy tomu tak opravdu je, chyba modelu je vyjádřena kombinací normálního a rovnoměrného rozdělení. Tato modifikace je plnohodnotným nahrazením RANSACu a ve většině případů dosahuje lepších výsledků.

PROSAC

PROSAC (*Progressive Sample Consensus*) vyžaduje od uživatele, aby určil rozložení chyb v datových souborech [13]. Pracuje totiž nejdříve s daty, které obsahují nejméně chyb. Z nich sestaví model a ten postupně rozšiřuje o data, která obsahují chyb více. Díky tomu se snižuje pravděpodobnost sestavení špatného základního modelu. Pokud uživatel zadá informace o chybách nesprávně nebo vůbec, PROSAC dosáhne nejhůře stejného výsledku jako RANSAC.

Princip PROSACu je vhodné implementovat i do jiných variant, například do výše zmíněných metod MLESAC a MSAC.

Randomized RANSAC

Dalším možným přístupem, jak docílit zrychlení původního algoritmu RANSAC, je Randomized RANSAC [14].

Každá iterace v rámci tohoto algoritmu má dva kroky. První krok probíhá podobně jako u RANSACu, ale je kontrolováno pouze d bodů z celkového počtu N . Pokud jsou všechny tyto body označeny jako inliery, algoritmus pokračuje druhým krokem, kdy je kontrolováno zbylých $N - d$ bodů. Poté se kontroluje celkový počet inlierů mezi N body a pokud je dostatečný, je pomocí nich zpřesněn výsledný model.

Rychlost algoritmu je závislá na relativní četnosti úspěšných prvních a druhých kroků. Ta lze ovlivnit vhodným výběrem parametru d . Vysoká hodnota d má za následek nízkou úspěšnost prvního kroku, neboť se zvyšuje pravděpodobnost výskytu outlieru. Naopak pokud se zde outlier nevyskytne, zvýší se pravděpodobnost úspěšnosti druhého kroku. Špatná volba tohoto parametru může způsobit, že tato metoda bude pomalejší než původní RANSAC.

3.3 Houghova transformace

Houghova transformace je pojmenována po svém autorovi P. V. C. Houghovi, který si ji roku 1962 nechal patentovat. Původní metoda sloužila k transformaci bodů v rovině na rovinné křivky. Později byla zobecněna dvojicí Richardem Dudou a Peterem Hartem [15].

Naším úkolem je transformace bodu v prostoru \mathbb{R}^3 . Transformace se bude lišit v závislosti na druhu hledaných ploch. V případě hledání rovin se jedná o transformaci z \mathbb{R}^3 do $\langle 0, 2\pi \rangle \times \langle 0, \pi \rangle \times \mathbb{R}$ [16]. Každý bod je zde reprezentován plochou a každá rovina je zobrazena jako dva body. Zobrazení bodu lze také popsat jako množinu všech zobrazení rovin, kterým bod náleží.

3.3.1 Transformace bodu pro detekci rovin

Každou rovinu v \mathbb{R}^3 lze popsat pomocí normálového vektoru $n = (n_x, n_y, n_z)$ a jednoho příslušného bodu $P = [P_x, P_y, P_z]$. Další závislou vlastností je její vzdálenost ρ od počátku souřadného systému.

Vzdálenost ε tohoto bodu P od vlastní roviny je

$$\varepsilon = \frac{P_x \cdot n_x + P_y \cdot n_y + P_z \cdot n_z + D}{\sqrt{n_x^2 + n_y^2 + n_z^2}} = 0. \quad (3.7)$$

Protože pro jednotkový vektor platí

$$\|n\| = \sqrt{n_x^2 + n_y^2 + n_z^2} = 1, \quad (3.8)$$

dostáváme po úpravě

$$-D = P_x \cdot n_x + P_y \cdot n_y + P_z \cdot n_z. \quad (3.9)$$

Po určení D můžeme vyjádřit vzdálenost ρ této roviny od počátku $O = [0, 0, 0]$, přičemž budeme uvažovat $D' = -D$,

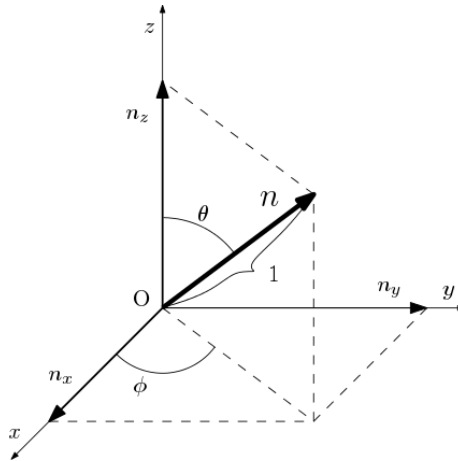
$$\rho = \frac{0 \cdot n_x + 0 \cdot n_y + 0 \cdot n_z + D'}{\sqrt{n_x^2 + n_y^2 + n_z^2}} = 0. \quad (3.10)$$

Po opětovné úpravě a dosazení vztahů (3.8) dostáváme

$$\rho = P_x \cdot n_x + P_y \cdot n_y + P_z \cdot n_z. \quad (3.11)$$

Složky jednotkového normálového vektoru si vyjádříme ve sférických souřadnicích, přičemž θ je úhel, který svírá normálový vektor s rovinou XY , a φ je úhel mezi normálovým vektorem a osou z (obr. 5).

$$\begin{aligned} n_x &= \sin \theta \cdot \sin \varphi \\ n_y &= \cos \theta \cdot \sin \varphi \\ n_z &= \cos \varphi, \end{aligned} \quad (3.12)$$



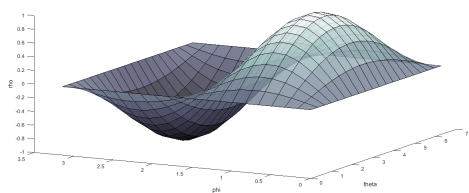
Obrázek 5: Sférické souřadnice jednotkového normálového vektoru, $\rho = 1$

Po dosazení vztahů (3.12) do (3.11) dostáváme

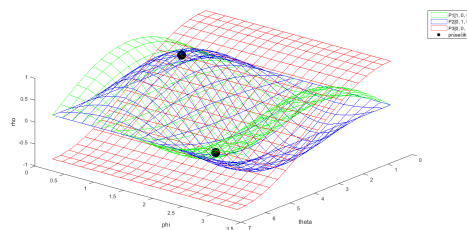
$$\rho = P_x \cdot \sin \theta \cdot \sin \varphi + P_y \cdot \cos \theta \cdot \sin \varphi + P_z \cdot \cos \varphi. \quad (3.13)$$

Vidíme, že pro daný bod P je vzdálenost ρ jednoznačně dána úhly θ a ϕ . Trojice θ, ϕ, ρ definuje Houghův prostor (θ, ϕ, ρ) . Pro zobrazení bodu v Houghově prostoru je nutné, abychom každé dvojici $(\theta, \phi) \in \langle 0, 2\pi \rangle \times \langle 0, \pi \rangle$ určili příslušné ρ podle (3.13). Vznikne tak plocha, jež je zobrazením bodu P (obr. 6).

Spolu s bodem P oba vektory $n_1 = (n_x, n_y, n_z)$ a $n_2 = (-n_x, -n_y, -n_z)$ určují totožnou rovinu, ale v Houghově prostoru se tato rovina zobrazí jako dva body; každému vektoru bude odpovídat jiný bod (obr. 7). Proto nám stačí uvažovat θ pouze na intervalu $\langle 0, \pi \rangle$.



Obrázek 6: Zobrazení bodu v Houghově prostoru

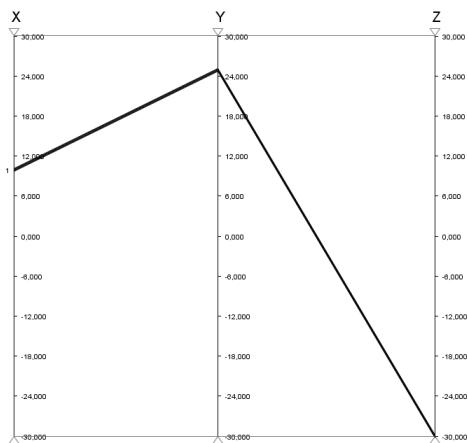


Obrázek 7: Zobrazení tří bodů se protínají ve dvou bodech

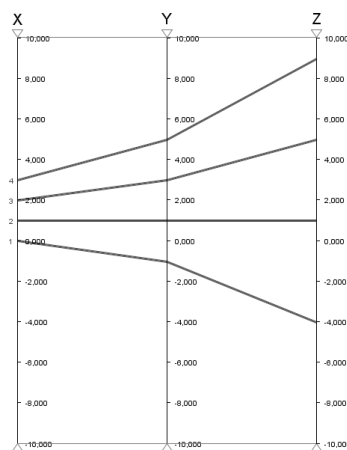
3.4 Paralelní souřadnice

Popsání této metody je od roku 1885 často připisováno francouzskému matematikovi a inženýrovi Philbertu Maurice d'Ocagnovi, nicméně nesprávně. Ten sice použil toto slovní spojení, ale činil tak v souvislosti s nomogramem. Člověk, který významně rozšířil paralelní souřadnice tak, jak jsou chápány v této kapitole, je Alfred Inselberg, který tuto metodu objevil v 50. letech minulého století [18].

Paralelní souřadnice se používají především k zobrazování multidimenzionálních dat. Pro zobrazení n -rozměrných dat je zapotřebí n rovnoběžných os, pro každý rozměr jedna. Bod je zobrazen jako lomená čára s vrcholy na jednotlivých osách. Poloha každého vrcholu na dané ose je určena příslušnou souřadnicí. Jedná se o transformaci z \mathbb{R}^n do \mathbb{R}^2 . My budeme uvažovat pouze prostor \mathbb{R}^3 . Na obr. 8 je zobrazen bod $P[10, 25, -30]$ v paralelních souřadnicích. Všechny grafy v této kapitole byly vytvořeny pomocí programu *XDAT*.



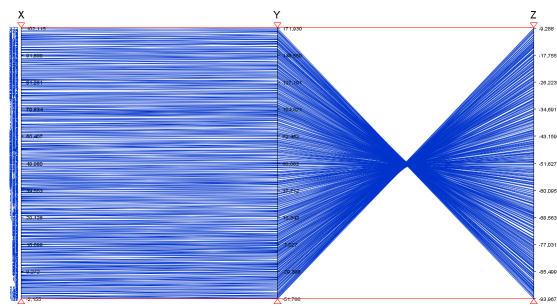
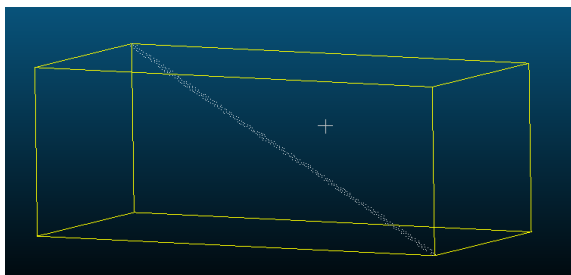
Obrázek 8: Bod $P[10, 25, -30]$ v paralelních souřadnicích



Obrázek 9: Zobrazení více bodů

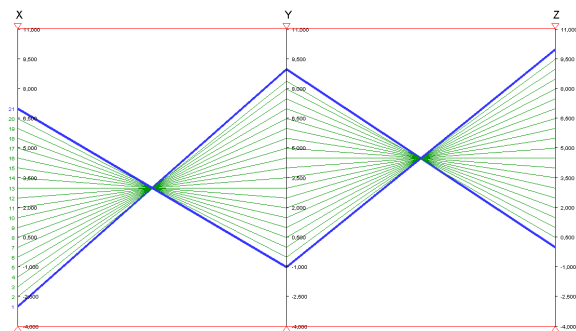
3.4.1 Vlastnosti

Pokud se lomené čáry protínají v každém úseku mezi osami nebo jsou v něm navzájem rovnoběžné, pak tyto body leží na jedné úsečce. Oba případy lze vidět na obr. 10, kde jsou zobrazeny body ležící podél úsečky. Jistým úskalím je skutečnost, že se lomené čáry nemusí v daném úseku protnout, ale prodloužení příslušných úseček mimo daný úsek ano, viz obr. 9.



Obrázek 10: Body v prostoru ležící v blízkosti stejné úsečky a jejich reprezentace v paralelních souřadnicích

Pro určení přímky z grafu je nutné nejprve najít průsečík více lomených čar. Dále je nutné vybrat nejméně dvě lomené čáry procházející tímto průsečíkem a určit z nich příslušné body. Těmito body je pak jednoznačně dána přímka. V případě reálných dat, ve kterých se mohou nacházet chyby, místo průsečíku uvažujeme jeho okolí a přímku je potřeba aproximovat pomocí většího počtu bodů. Místo přímky lze uvažovat i úsečku, jejíž krajní body ohraničují nalezenou strukturu v grafu (obr. 11). Poté lze jednoduše určit jejich souřadnice.



Obrázek 11: Modré čáry v grafu reprezentují okrajové body dané úsečky

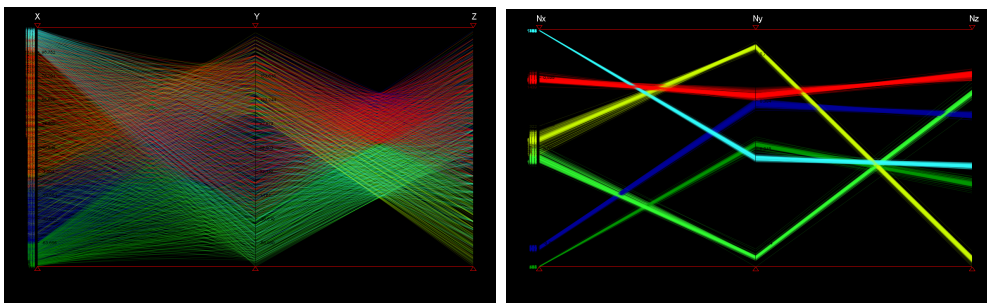
Při velkém množství bodů velice rychle dochází k znepráhlednění grafu. Pro lehké nalezení struktur pouhým pohledem se grafy upravují barevně [19]. Základní úpravou je použití více stupňů jedné barvy. Jsou tak zvýrazněna místa, kterými prochází více čar. Další úpravou může být vhodná změna barvy pozadí, případně převedení celého grafu do negativu.

3.4.2 Využití pro clustering

Při zpracování mraku bodů je většinou cílem nalézt plochy. Zatím nebyla popsána efektivní metoda používající paralelní zobrazení, která by byla schopna nalézt plochy. Existuje ale velice podobný způsob zobrazení, který je popsán v kapitole 3.4.3, který umožňuje například nalezení rovin. Nicméně i paralelní souřadnice lze za specifických podmínek použít.

Soubor s daty ze skeneru často obsahuje nejen souřadnice každého bodu, ale i jeho normálu. Právě pomocí transformace složek normál do paralelních souřadnic můžeme roz-

třídít body na jednotlivé roviny za předpokladu, že na snímaném předmětu se nenachází roviny se stejnou orientací normály. Výsledek lze vidět na obr. 12.

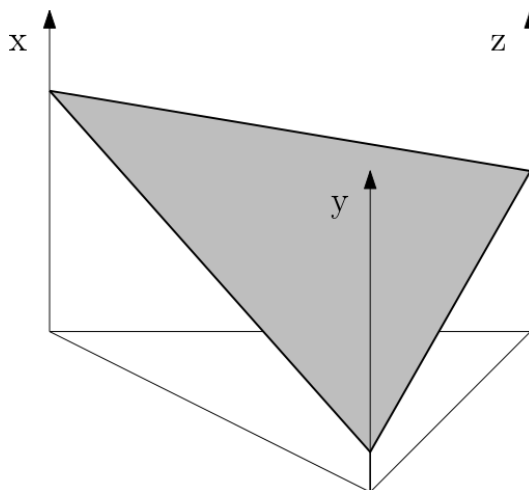


Obrázek 12: Body jsou označeny barevně podle příslušných rovin
Vlevo jsou paralelní souřadnice bodu, vpravo paralelní souřadnice normály bodu

3.4.3 Paralelní prostor

Paralelní prostor vznikne transformací z \mathbb{R}^3 do \mathbb{R}^3 tak, že jednotlivé souřadnice bodu se opět zobrazí na osy, ale tyto osy leží v jednotlivých rozích trojúhelníku a jsou kolmé na rovinu trojúhelníku. Každý bod je proto reprezentován rovinou, která protíná osy v místech odpovídajících souřadnicím (obr. 13).

Toto zobrazení umožňuje nalezení rovin [21]. Roviny, které jsou zobrazením bodů ležících ve stejné rovině, mají společný nejméně jeden bod. V některých speciálních případech je tento bod nevlastní (např. rovina $x - y = 0$). Můžeme tedy zpětně určit souřadnice tří takových bodů a jim přiřadit společnou rovinu. V případě chyb v měření je nutné postupovat podobně jako u paralelních souřadnic.



Obrázek 13: Zobrazením bodu v paralelním prostoru je rovina

3.5 Srovnání metod

Z hlediska clusteringu je v současnosti nejvíce používána metoda RANSAC a její modifikace. Tato metoda, na rozdíl od ostatních, je určena k sestavení modelů z dat, které

obsahují větší počet chyb. Protože se jedná o nedeterministický algoritmus, nemáme zaručen stejný výsledek při každém spuštění. Úspěch proto může záviset na náhodě. Naštěstí existuje mnoho vhodných modifikací, které při vhodném použití zvyšují šanci na úspěch.

Houghova transformace a paralelní prostor převádějí každý bod na plochu. Problémem tudíž zůstává, jak různým plochám určit společný bod, který představuje hledanou rovinu. Často používaným nástrojem je diskretizace oblasti prostoru (rozdělení oblasti na více menších oblastí). Pro každou plochu se zjistí, kterými oblastmi daná plocha prochází. Při dostatečně malém dělení určíme oblast, kterou prochází nejvíce ploch a poté lze například určit společný bod (hledanou rovinu) jako střed této oblasti. Nevýhodou použití paralelního prostoru je neohraničenost prostoru, ve kterém se může nacházet bod představující hledanou rovinu.

Tyto dvě metody ovšem neuvažují chyby v měření. Často je potřeba uvažovat větší okolí bodu, než jsou uvažované podoblasti. Proto se například Houghova transformace používá v kombinaci s RANSACem.

Všechny možnosti paralelního prostoru ještě nebyly objeveny, a proto je v současnosti vhodný pouze pro hledání rovin. Houghova transformace je závislá na nalezení vhodné transformace bodu na plochu. Naopak metoda RANSAC nevyžaduje žádnou transformaci a je schopná pracovat s explicitními, implicitními i parametrickými rovnicemi.

Protože principy metod jsou poměrně odlišné, je vhodné využívat pozitivních vlastností jednotlivých metod a metody vhodným způsobem kombinovat.

4 Implementace RANSACu

4.1 Cíle implementace

Hlavním cílem implementace bylo vytvořit funkční program, který pomocí metody RANSAC vyhledá roviny v zadaném mraku bodů. Vstupní parametry vyžadované zmíněnou metodou měl vkládat uživatel. Výstupem měly být nalezené roviny v podobě parametrických rovnic i jejich grafické zobrazení.

Během programování hlavního algoritmu vyvstala otázka, zda by nebylo vhodné se získanými daty dále pracovat a ověřit tak obecně možnost jejich dalšího využití. Algoritmus RANSAC je často používán k hledání střech. Mrak bodů použitý při testování byl proto získán skenováním zmenšeného modelu domu. Vzhledem k této skutečnosti bylo dalším cílem určit ty nalezené roviny, které by mohly představovat střechy a dále určit jejich vhodnost pro umístění solárních panelů. Proto je součástí programu i solární kalkulačka, která zahrnuje více vstupních údajů než většina základních kalkulaček, které je možné dohledat na internetu. Nicméně neumí pracovat s dostupnými meteorologickými daty jako více sofistikované programy. Solární kalkulačka je věnována samostatná kapitola 5.

4.2 Způsob implementace

Program byl nazván CMB, což je zkratka pro clustering mraku bodů – název zadání práce. Zvoleným programovacím jazykem byl C#. Použito bylo vývojové prostředí Microsoft Visual Studio 2015.

Pro program bylo nutné vytvořit dva formuláře a sedm vlastních tříd. Algoritmus RANSACu se nachází v hlavní třídě Main.cs, která využívá ostatní třídy. Dalšími třídami jsou Loader.cs, Map.cs, Plane.cs, Point.cs, Roof.cs, SolarCalculator.cs a Vector.cs. Do programu byla přidána třída gnuplot.cs dostupná z GitHubu [22], která umožňuje 3D vizualizaci objektu pomocí programu gnuplot. Tato třída byla pro účely vlastního programu mírně upravena, aby bylo možné gnuplot v případě chyby spustit znovu.

Gnuplot je zdarma dostupný program, který umožňuje plotování skrze příkazový řádek. Ke správnému fungování v rámci programu je potřeba nastavit cestu k programu v panelu Nastavení.

Zdrojový kód celého programu je přiložen v příloze.

4.3 Rozdíl mezi implemantací a původním algoritmem

V některých úpravách RANSACu se ze všech iterací vybere výsledek té iterace, která k sestavení modelu vybrala nejvíce bodů. Během testování programu se ukázalo, že vzhledem k nedokonalé rovnému povrchu tento postup nezaručuje nejpřesnější řešení.

Vnitřní algoritmus byl dále upraven tak, aby mohl nalézt více rovin během jednoho spuštění. Navíc se při každé iteraci neuvažují body, které již byly přiřazeny dříve nalezené rovině. Při vhodně zvolených vstupních hodnotách byl výsledný model dostatečně přesný a navíc se celá procedura významně zrychlila. Uživatel je nabídnuta i 3D vizualizace výsledku, která umožňuje snadno identifikovat případnou chybu.

4.4 Vstupy

Mezi vstupy hlavní části programu je cesta k souboru s naměřenými body. Tento soubor musí být ve formátu textového souboru a musí obsahovat pouze body, které jsou jednotlivě na řádcích zapsány ve formátu $X; Y; Z;$. Protože většina dostupného softwaru, ze kterého lze takovýto soubor vytvořit, má anglické prostředí, je místo desetinné čárky požadována tečka.

Dále je nutné, aby uživatel zadal vstupní parametry pro metodu RANSAC. Prvním požadovaným parametrem je *threshold*. Fyzikální jednotka odpovídá jednotce dat v textovém souboru. Dalším požadovaným parametrem je počet inlierů. Protože se předpokládá rozdíl v plošných obsazích jednotlivých ploch, může uživatel zadat parametry pro více ploch a oddělit je středníkem. Pokud uživatel zná dopředu počet rovin na tělese, může tento údaj také zadat. Pokud ne, vyplní v příslušné kolonce nulu. Posledním údajem v této části je maximální počet iterací. Obecně platí, že čím více iterací, tím se zvyšuje šance na nalezení rovin, ale také se zvyšuje potřebný čas.

4.5 Výstupy

Mezi výstupy patří přehled nalezených rovin za použití implicitních rovnic. Dále je u každé roviny uveden počet bodů a pořadí iterace, v níž byla daná rovina nalezena. Uživatel má možnost grafického vykreslení za použití programu *gnuplot* (obr. 15).

Vlastnosti nalezených rovin jsou dále zpracovány a používány solární kalkulačkou.

4.6 Popis algoritmu

Nejprve je spuštěn algoritmus RANSAC. Ten v každé iteraci zvolí náhodně tři body a proloží jimi rovinu. Poté hledá všechny body, které leží v menší vzdálenosti od této roviny než je zadaný *threshold*. Pokud je počet těchto inlierů větší než je požadováno, je plocha uložena a inliery jsou odebrány z původní množiny. Implicitní rovnice je zpřesněna použitím metody nejmenších čtverců. Při hledání další plochy je zohledněn požadavek na možné jiné procentuální zastoupení inlierů. Po kontrole počtu inlierů je iterace ukončena. Iterace probíhají, dokud není nalezen požadovaný počet ploch nebo dokud není dosaženo maximálního počtu iterací.

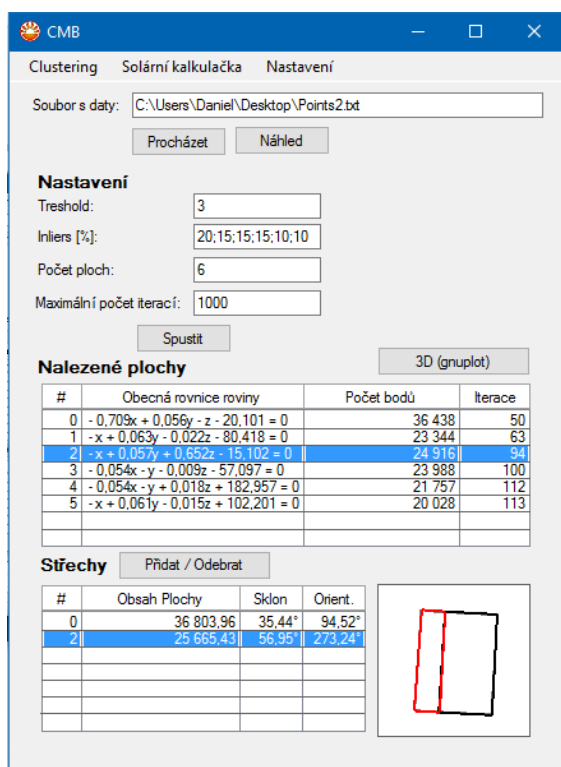
Následně je zjištěno, které roviny by mohly být považovány za střechu. To je určeno výpočtem úhlu mezi normálou roviny a osou z . U nalezených střech je spočítán obsah plochy za pomoci metody Andrew's monotone chain (kapitola 2.2.2) a Gausovy metody (kapitola 2.2.3). Před jejich použitím je potřeba otočit každou rovinu tak, aby byla rovnoběžná s rovinou XY (kapitola 2.2.1), což umožní používat pouze dvě souřadnice namísto tří, jak je těmito metodami vyžadováno.

4.7 Grafické rozhraní a práce s programem

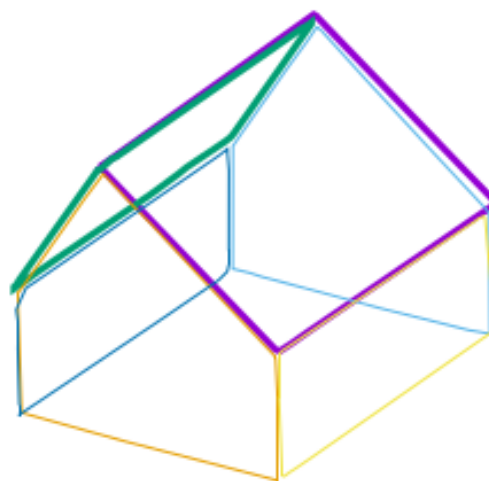
Veškeré vstupy a výstupy pro část programu zabývající se clusteringem se nachází na jednom panelu. Další panely jsou věnovány solární kalkulačce a nastavením.

Uživatel zde má možnost zadat cestu k souboru s daty. Po zadání cesty k souboru má možnost rychlého náhledu na nahrané body (pro rychlost a plynulost zobrazení se nezobrazí všechny). Pak má uživatel možnost zadat vstupní údaje pro algoritmus RANSACu a spustit jej.

Po dokončení algoritmu se uživateli vypíše nalezené roviny a dodatečné informace o rovinách určených jako střechy. Pokud uživatel zvolí libovolnou rovinu, je zvýrazněn příslušný konvexní n-úhelník v půdorysovém náhledu. Dále má uživatel možnost přidat nebo odebrat rovinu ze seznamu střech.



Obrázek 14: Panel RANSACu



Obrázek 15: Grafický výstup pomocí gnuplotu

4.8 Testování na reálných datech

Testování probíhalo na dvou různých typech dat. První sada data byla získána pomocí skeneru ATOS naskenováním modelu. Na těchto datech se testovala především rychlost vnitřního algoritmu. Další data pocházela z leteckého skenování a byla získána z internetu. Data z leteckého snímání obecně nemají takovou hustotu bodů jako první typ. Navíc body nepokrývají celou oblast rovnoměrně. Program byl proto testován na zpracování i takového typu dat.

Program byl testován na počítači Toshiba Satellite L850 s 64 bitovým operačním systémem Windows 10. Počítač byl vybaven operační pamětí 4GB RAM, dvoujádrovým procesorem Intel Core i5 a diskem HDD s 5400 otáčkami. Stáří počítače v době testování bylo 4 roky, což může mít negativní vliv na stav uvedených počítačových součástí.

Data pro první testování byla získána pomocí skenování papírového modelu. Během měření byl použit skener ATOS CompactScan od německé firmy GOM. Spolu se skenerem byl použit software ATOS Professional verze V.7.5 SR2, který sbírá data z jednotlivých skenů a sestavuje z nich jeden soubor, přičemž hlídá vzájemnou polohu bodů z různých skenů. To je umožněno pomocí referenčních bodů. Při tomto měření měly podobu černo-bílých terčíků s průměrem 3 mm, kterými byl papírový model polepen před samotným měřením.

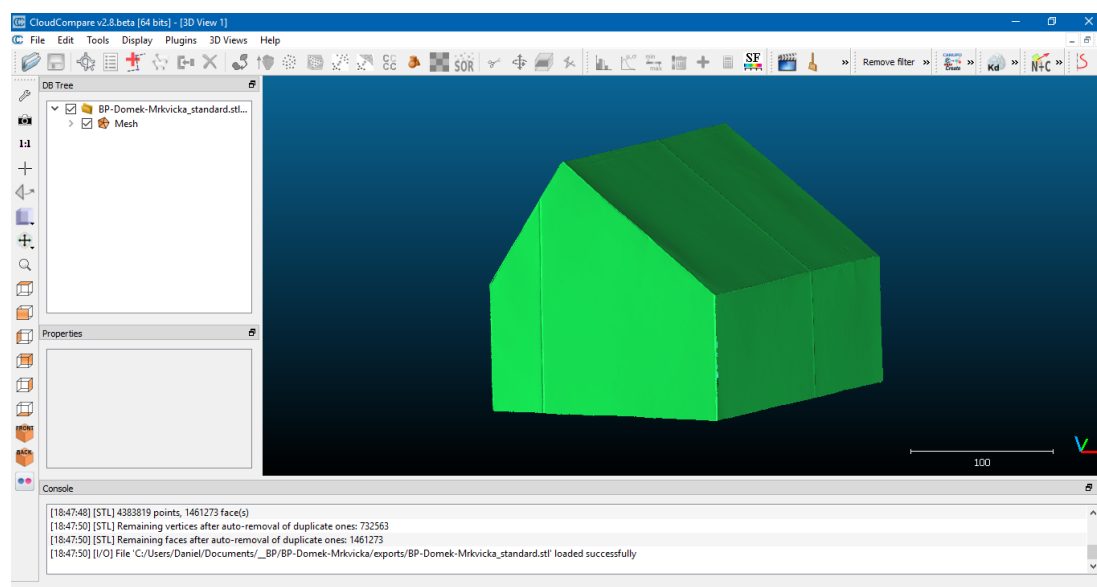
Výstupem měření byl mesh soubor, v němž je naskenovaný objekt uložen jako mnohostěn, přičemž jeho stěny byly dostatečně malé vzhledem k rozměrům modelu. Tento



Obrázek 16: Fotografie z měření

soubor byl dále zpracován pomocí opensource softwaru CloudCompare verze v2.8.beta. V něm bylo provedeno otočení modelu a vygenerování textového souboru s požadovaným počtem bodů. Tento soubor obsahuje na každém řádku souřadnice jednoho bodu. Jednotlivé souřadnice jsou odděleny pomocí středníku. Takový soubor už lze otevřít v programu, který je součástí této bakalářské práce.

Software CloudCompare lze využít i k úpravě dat volně dostupných ke stažení na internetu. Umí otevřít velké množství souborových formátů a nabízí mnoho možností úprav. Proto byl tento software použit také k úpravě dat z leteckého skenování.



Obrázek 17: Prostředí programu CloudCompare

Program byl testován z hlediska časové náročnosti a počtu potřebných iterací. Byly použity tři testovací mraky bodů, které obsahovaly nasnímané body stejného objektu, ale pokaždé s jinou hustotou. Testovací mraky měly přibližně sto tisíc, jeden milión a deset miliónů bodů. Pokaždé byl celý proces spuštěn dvacetkrát. Sledován byl čas potřebný pouze pro algoritmus RANSACu a celkový počet iterací. Výsledky měření jsou uvedeny v Tabulce 1. Krajní hodnoty jsou zvýrazněny tučně. Lze vidět velký rozsah naměřených hodnot. Program je oproti původnímu očekávání rychlý, jeden milión bodů byl schopen vyhodnotit ve většině případů do jedné minuty.

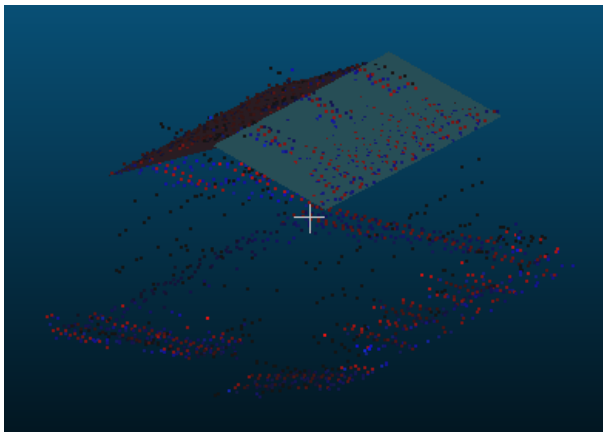
Ostatní operace s body, jako je například jejich načítání z textového souboru nebo určování okrajových bodů konvexního n-úhelníku pro každou rovinu, činí při deseti milio-

nech bodů další dvě minuty potřebného času, při jednom milionu bodů se přibližně jedná o deset vteřin a při stech tisících bodů lze tento čas plně zanedbat.

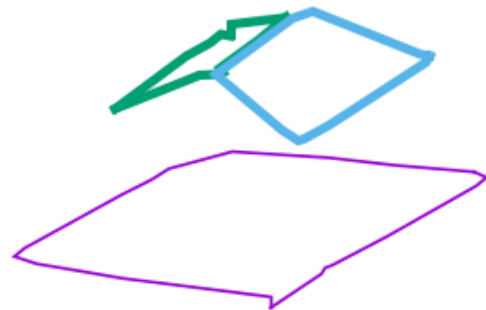
Tabulka 1: Výsledky měření

počet bodů	100 000		999 811		9 999 407	
č. měření	čas [ms]	iterace	čas [ms]	iterace	čas [ms]	iterace
1	2 799	215	16 874	148	307 129	275
2	460	48	69 896	408	201 882	147
3	2 011	261	13 640	131	358 383	319
4	1 386	132	3 453	102	18 466	22
5	598	54	50 870	455	327 729	274
6	1 703	141	8 415	92	49 460	52
7	1 878	201	22 084	195	262 639	205
8	851	99	18 676	180	140 306	121
9	1 452	129	23 749	241	51 044	62
10	1 995	187	59 758	524	175 800	174
11	1 834	151	31 000	271	155 797	178
12	1 748	160	14 839	100	109 712	161
13	2 883	222	6 199	60	142 141	138
14	1 316	123	22 035	233	217 377	185
15	2 314	216	23 195	233	121 802	127
16	4 118	340	14 233	127	298 946	252
17	773	103	24 647	197	83 216	91
18	3 308	258	18 654	195	157 272	124
19	1 383	112	5 497	67	200 720	205
20	2 083	193	16 689	129	144 185	133
prům. hodnoty	1845	167	23 221	201	176 200	162

Další testování se týkalo schopnosti pracovat s daty z leteckého snímání. Byl použit soubor MT73_08_au.las dostupný na internetových stránkách španělské provincie Gipuzkoa [23].



Obrázek 18: Letecký sken v CloudCompare



Obrázek 19: Letecký sken v CMB

V tomto souboru se nachází sken čtvercové oblasti zahrnující moře, plážové městečko a jeho okolí. Konkrétně se jedná o obci Zarautz ve Španělsku. Body nepokrývají celou

plochu, ale jsou rozmístěny v od sebe oddělených pásech. Protože data byla pořízena pouze shora, je počet bodů bočních stěn nedostatečný. Proto lze předpokládat, že nebudou nalezeny. Ze souboru byla odstraněna přebytečná data, tak aby zůstaly body pouze jednoho vybraného domu. Soubor byl zpracován pomocí programu CloudCompare, který mimo jiné zahrnuje plugin RANSACu. Tohoto pluginu bylo využito ke srovnání výstupu specializovaného optimalizovaného softwaru s vlastním programem. U jiných programů se nepodařilo dohledat popis používaných algoritmů, a proto nemohly být testovány a srovnávány s vlastním programem.

Nejtěžší bylo odhadnout správné vstupní parametry. Po srovnání obr. 18 a obr. 19 lze konstatovat, že program je schopen správně pracovat i s daty leteckých skenů.

5 Solární kalkulačka

5.1 Teorie

Po detekci a určení rovinných ploch program přechází do druhé fáze, která je založena na určení solárního potenciálu dané střechy při jejím osazení solárními panely. Předpokladem je, že solární panely dokonale pokrývají celou střechu jako na obr. 20. Pro jiné případy je uváděn i přepočet výsledných hodnot na jeden metr čtvereční. Množství slunečního záření je vypočteno aplikací vzorců ze stránek Photovoltaic Education [24]. V Tabulce 2 jsou uvedeny použité fyzikální veličiny včetně jednotek.

Tabulka 2: Přehled použitých symbolů

veličina	název	jednotka
ϕ	zeměpisná šířka	$[\circ]$
θ	zeměpisná délka	$[\circ]$
δ	deklinace	$[\circ]$
α	výška	$[\circ]$
ζ	zenitová vzdálenost	$[\circ]$
β	azimut	$[\circ]$
μ	sklon solárního panelu	$[\circ]$
ν	natočení solárního panelu	$[\circ]$
d	den v roce	$[-]$
h	nadmořská výška	$[km]$
S	obsah plochy solárního panelu	$[kWh]$
EoT	časová rovnice	$[min]$
LT	místní standardní čas	$[h]$
LST	místní sluneční čas	$[h]$
HRA	hodinový úhel	$[\circ]$
AM	koeficient vzduchové hmoty	$[-]$
I_D	přímé sluneční záření	$[kWm^{-2}]$
I_G	celkové sluneční záření	$[kWm^{-2}]$
S_{module}	sluneční záření dopadající na solární panel	$[kW]$

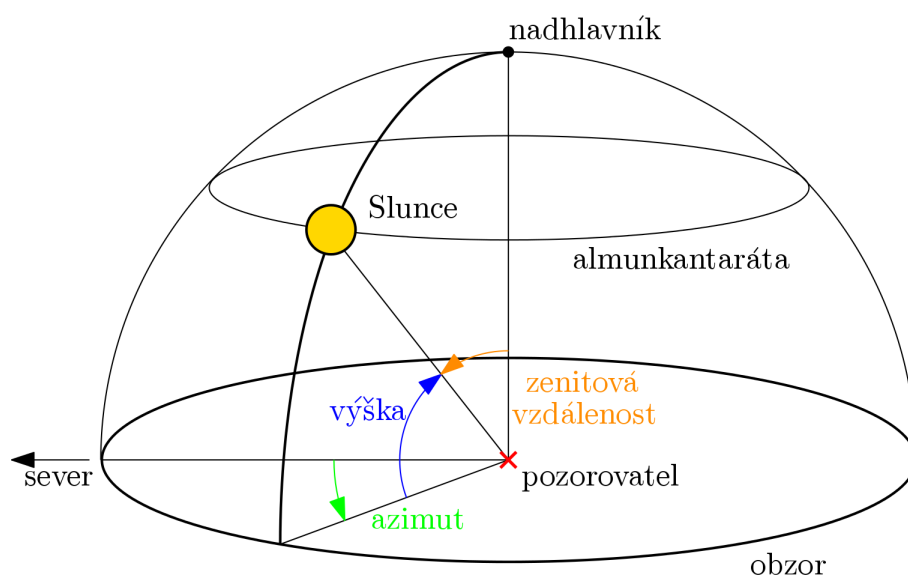
Model umožňuje spočítat celkové sluneční záření S_D dopadající v libovolný okamžik na rovný solární panel. Využívá k tomu popis relativního pohybu Slunce vůči Zemi. Pozice Slunce vůči pozorovateli na Zemi je popsána pomocí výšky a azimutu, viz obr. 21. Zásadní věc, kterou model zanedbává, je vliv počasí. Na účinnost panelu mají vliv faktory jako teplota, vítr a oblačnost. Vzhledem k rozdílným klimatickým podmínkám po celé Zemi a proměnlivosti počasí během roku, by po uživateli programu byla požadována rozsáhlá data. Hodnoty získané pomocí programu budou proto vyšší než hodnoty získané reálným měřením.

Vstupními daty pro model jsou zeměpisná šířka ϕ a délka θ , nadmořská výška h , den roku d , místní standardní čas LT a vlastnosti panelu, kterými jsou obsah plochy S , sklon μ a natočení ν vzhledem k severu (pro panel natočený k severu $\nu = 0^\circ$, k východu $\nu = 90^\circ$, ...).

Prvním pomocným údajem, který můžeme spočítat, je deklinace δ . Deklinace je pozice Slunce vůči rovině rovníku. Ta se mění během celého roku, ale její změna během dne je



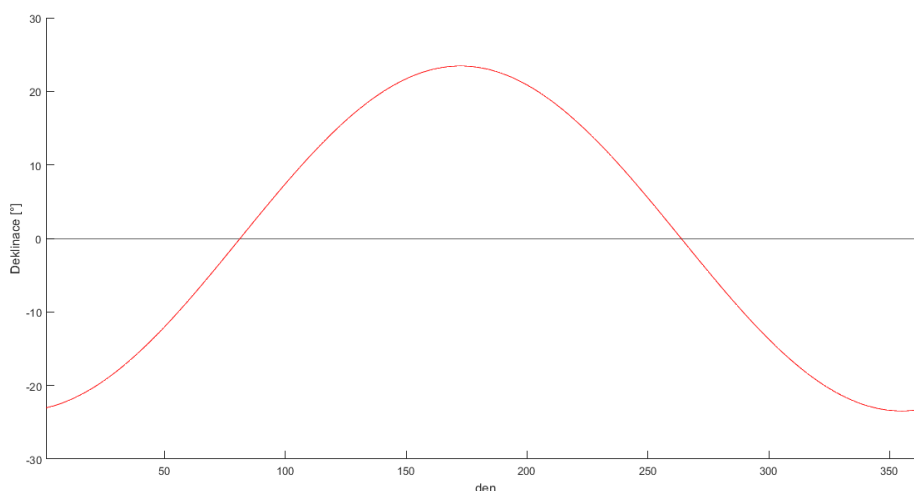
Obrázek 20: Střecha pokrytá solárními panely
CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=842696>



Obrázek 21: Astronomický popis pozice Slunce

nepatrná. Je způsobena zkosením zemské osy o $23,45^\circ$. Proto deklinace nabývá hodnot mezi $-23,45^\circ$ a $23,45^\circ$. Krajních hodnot nabývá během slunovratů (obr. 22) a Slunce se v tu dobu nachází nad jedním z obratníků.

$$\delta = -23,45^\circ \cdot \cos\left(\frac{360}{365} \cdot (d + 10)\right)$$



Obrázek 22: Graf deklinace v závislosti na dnu v roce

Dalším údajem, který je potřeba získat, je hodinový úhel HRA . Hodinový úhel je úhel mezi místním poledníkem a deklinační kružnicí Slunce. Deklinační kružnice je kružnice procházející oběma zemskými póly (obdoba poledníku).

K určení hodinového úhlu je nutné znát pravý sluneční čas LST . Existují dva sluneční časy – střední a pravý. Střední sluneční čas předpokládá rovnoměrný pohyb Slunce a odpovídá našemu měření času. To znamená, že každý den má 24 hodin a Slunce je nejvýše na obloze v pravé poledne. Země ale obíhá Slunce po těsně eliptické dráze, což způsobuje nerovnoměrný pohyb Slunce, který je popsán pravým slunečním časem. Rozdíl mezi pravým a středním slunečním časem se během roku mění a je popsán pomocí časové rovnice EoT . Je třeba zdůraznit, že časová rovnice je proměnná, nikoli rovnice. Její graf lze vidět na obr. 23.

$$EoT = 9,87 \sin(2B) - 7,53 \cos(B) - 1,5 \sin(B);$$

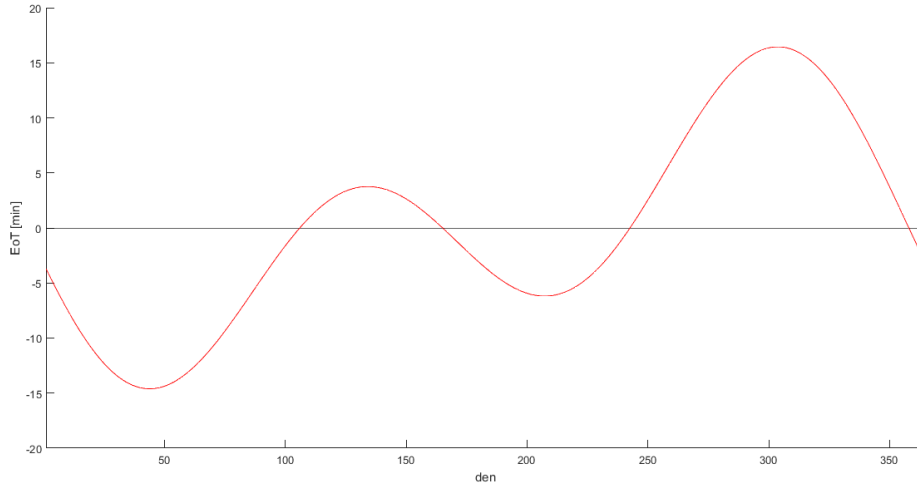
$$B = \frac{360}{365} \cdot (d - 81)$$

Po určení časové rovnice pro daný den můžeme určit pravý solární čas.

$$LST = LT + \frac{EoT}{60}$$

Převod pravého solárního času na hodinový úhel je velice jednoduchý. Jedné hodině odpovídá patnáct hodinových stupňů a 12 hodin je rovno 0° .

$$HRA = 15^\circ \cdot (LST - 12)$$



Obrázek 23: Graf časové rovnice v závislosti na dnu v roce

Všechny kružnice rovnoběžné s obzorem se nazývají almukantaráty. Výška α je úhlová vzdálenost almukantaráty, procházející tělesem, od obzoru. Jejím doplňkem do 90° je zenitová vzdálenost ζ .

$$\alpha = \sin^{-1}(\sin \delta \cdot \sin \phi + \cos \delta \cdot \cos \phi \cdot \cos(HRA))$$

$$\zeta = 90^\circ - \alpha;$$

Celou Zemi obklopuje atmosféra. V závislosti na výšce Slunce, respektive zenitovou vzdálenost, musí sluneční světlo putovat skrz ni různě dlouhou vzdálenost. Čím větší vzdálenost takto urazí, tím více je pohlceno sluneční energie. Tuto skutečnost popisuje koeficient vzduchové masy AM .

$$AM = \frac{1}{\cos \zeta + 0,505272 \cdot (96,07995 - \zeta)^{-1,6364}}$$

Přímé sluneční záření I_D je množství sluneční energie, která dopadne na 1 m^2 plochy. V případě přímého slunečního záření uvažujeme povrch vždy kolmý na dopadající záření. Konstanta 1,353 se nazývá solární konstanta. Je to množství záření před vstupem do atmosféry.

$$I_D = 1,353 \cdot 0,7^{AM^{0,678}}$$

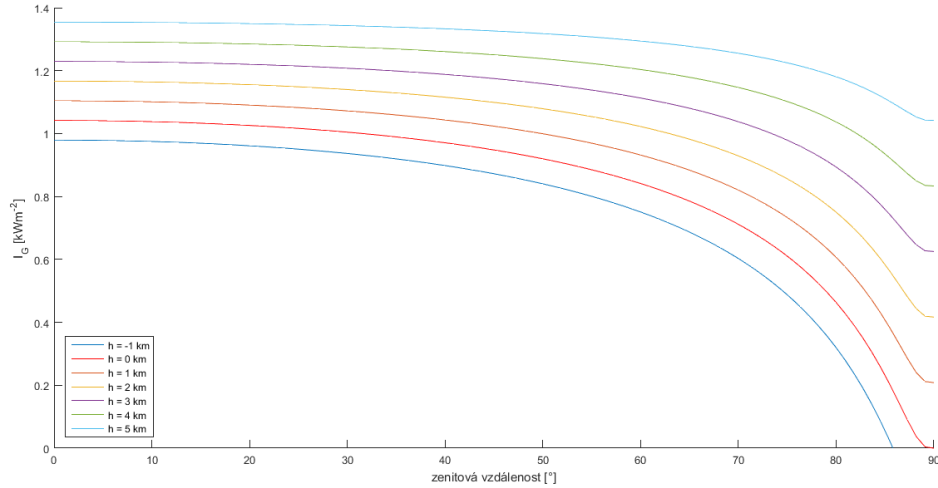
Na množství slunečního záření má vliv i nadmořská výška h . S rostoucí nadmořskou výškou záření přibývá. Při srovnání se skutečnými měřeními byl předchozí vzorec upraven, aby lépe odpovídal skutečnosti.

$$I_D = 1,353 \cdot ((1 - 0,14h) \cdot 0,7^{AM^{0,678}} + 0,14h)$$

Na obr. 24 lze vidět, že při velké zenitové vzdálenosti dochází k chybám, neboť jednotlivé křivky by se měly v bodě 90° (východ a západ Slunce) blížit k nule. Tuto chybu lze tolerovat, neboť výstupem programu jsou pouze celkové denní a roční hodnoty záření.

Kromě přímého slunečního záření dopadá na povrch i záření difúzní. Model předpokládá, že je rovno 10% přímého záření. Dostáváme tak celkové sluneční záření I_G .

$$I_G = 1,1 \cdot I_D$$



Obrázek 24: Graf celkového slunečního záření v závislosti na zenitové vzdálenosti a nadmořské výšce

K přesnému určení pozice Slunce na obloze je mimo výšky potřeba i azimut β . Společně tvoří takzvané obzorníkové souřadnice. Azimut je úhel mezi severem a průmětem tělesa na obzor. Je počítán kladně od severu směrem k západu.

$$\beta_0 = \frac{\sin \delta \cdot \cos \phi - \cos \delta \cdot \sin \phi \cdot \cos(HRA)}{\cos \alpha}$$

$$\beta = \begin{cases} \beta_0, & HRA < 0 \\ 360^\circ - \beta_0, & HRA > 0 \end{cases}$$

Nakonec se určí velikost slunečního záření dopadajícího na panel. V prvním kroku budeme předpokládat, že panel bude vždy kolmo na dopadající paprsky. Zajímá nás tak pouze velikost plochy.

$$S_{incident} = S \cdot I_D$$

Poté se uvažuje i jeho natočení a sklon. Sluneční záření S_{module} dopadající na solární panel je určeno součinem $S_{incident}$ a skalárního součinu vektoru slunečních paprsků a normály roviny panelu.

$$S_{module} = S_{incident} \cdot (\cos \alpha \cdot \sin \mu \cdot \cos(\nu - \beta) + \sin \alpha \cdot \cos \mu);$$

Při výpočtu nás zajímá pouze čas, kdy sluneční paprsky dopadají na panel, tedy doba mezi západem a východem Slunce. Tyto dva časové údaje lze jednoduše spočítat ze zeměpisné délky a deklinace s přihlédnutím k časové rovnici.

$$t_{východ} = 12 - \frac{1}{15^\circ} \cdot \cos^{-1}(\tan \phi \cdot \tan \delta) - \frac{EoT}{60};$$

$$t_{západ} = 12 + \frac{1}{15^\circ} \cdot \cos^{-1}(\tan \phi \cdot \tan \delta) - \frac{EoT}{60};$$

Tyto dvě hodnoty pomáhají zpřesnit model. Pokud je zadán čas v daný den mimo interval $\langle t_{východ}; t_{západ} \rangle$, pak je $S_{incident}$ vždy nutné nastavit roven nule. Obecně platí, že pokud bude $S_{incident}$ menší než nula, pak je nutné hodnotu interpretovat jako nulu.

5.2 Panel solární kalkulačky

Solární kalkulačka má v rámci programu svůj vlastní panel, jak je vidět na obr. 25. Uživatel zadává polohu objektu na mapě nebo přímým zadáním souřadnic. Dále nastavuje nadmořskou výšku a natočení celého domu. Dále volí použitou jednotku délky při měření. Informace o nalezených střechách se nacházejí v Tabulce 3. Všechny hodnoty se po změně libovolného vstupu ihned přepočítávají.

Po zvolení libovolné střechy lze zobrazit graf závislosti záření na dnu v roce, viz obr. 27. Červená křivka je hodnota pro zadaný sklon a natočení. Černá křivka pak ukazuje hodnoty pro panel, jehož normála neustále směřuje ke Slunci. Je tak možné srovnat efektivnost orientace daného panelu. V grafu jsou vyznačena data rovnodenností i slunovratů. Jedná se o významné body, ve kterých křivky nejčastěji nabývají maximálních nebo minimálních hodnot.

Použitý model umožňuje spočítat hodnotu záření pro libovolná reálná vstupní data. V programu se určují hodnoty celkového denního záření. Je tedy nutné nejprve určit čas východu a západu Slunce pro každý den. Mezi nimi se pak s krokem $\Delta h = 0,125$ hod napočítá celkové množství záření v jednotlivých okamžicích pro zadaný panel a pomocí numerické integrace se poté určí celkové denní hodnoty jak pro zadaný obsah plochy, tak obecně pro jeden metr čtvereční.

5.3 Způsob vyhodnocení získaných dat

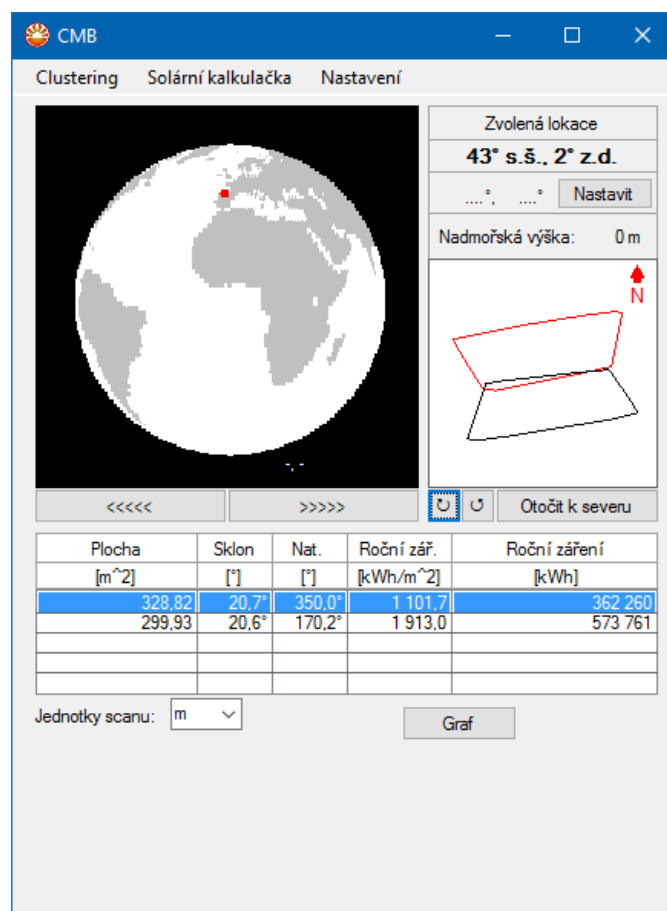
Solární kalkulačka a funkce na ni navazující byly testovány na datech z leteckého snímání (viz kapitola 4.8). Při zkoušení různých vstupních údajů se podařilo nalézt nejvýše dvě plochy střech z celkových čtyř. Obě tvarově odpovídaly skutečnosti. Jejich odhadovaná skutečná plocha má cca 290 m². Program určil jejich velikost na 299,93 a 328,82 m². V tomto případě došlo k relativní chybě 13,4 %. Na hřebenu střechy se nacházejí ploché panely (jak lze vidět na obr. 26), které nejspíše způsobily, že nalezené střechy se v tomto místě mírně překrývají a zvětšuje se tak velikost jejich ploch. Vlastní přesnost skenování se nepodařilo dohledat, ale lze předpokládat, že nebude zanedbatelná.

Po otevření panelu solární kalkulačky se provedlo nastavení lokace domu a jeho natočení vůči severu (obr. 25). Nakonec se nastaví použitá jednotka souboru s daty na jeden metr.

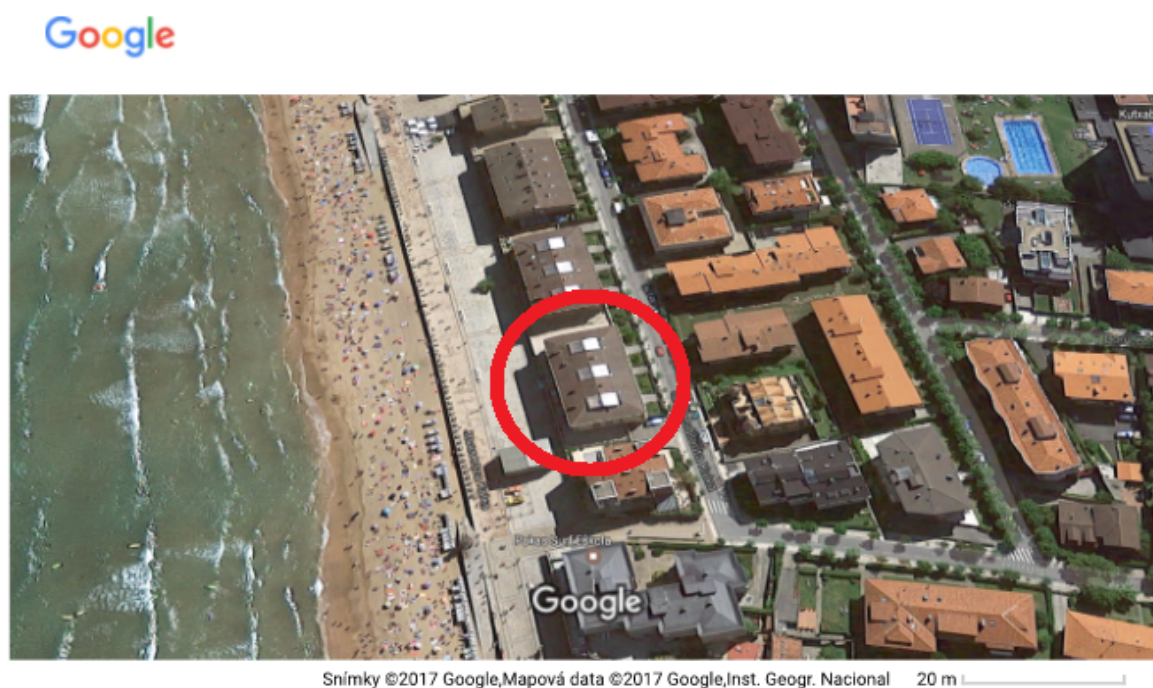
Získané výsledky (Tabulka 3) ukazují, že jižní střecha může teoreticky vyprodukovat až o 73 % více energie na metr čtvereční než střecha orientovaná k severu. Navíc při srovnání grafů obou střech (obr. 27, obr. 28) lze vidět, že produkce je u ní během roku více vyrovnaná. U severní střechy lze během zatažených dnů v zimě dokonce očekávat nulovou produkci. Primárně by se tedy solární panely měly instalovat na střechu jižní, což je obecné doporučení všech výrobců solárních panelů.

Tabulka 3: Výsledná data

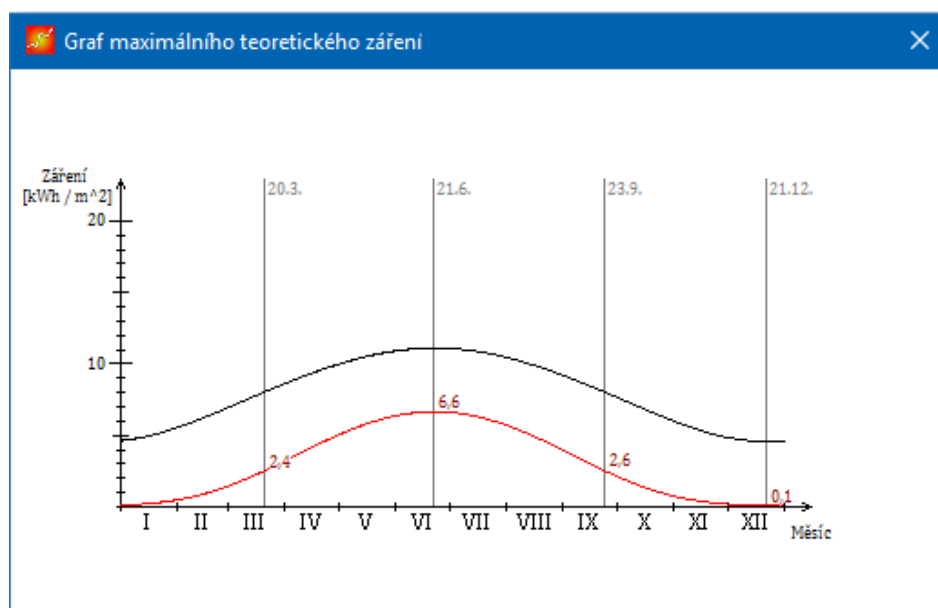
střecha	plocha [m ²]	sklon [°]	natočení [°]	roční záření [kWhm ⁻²]	roční záření [kWh]
severní	328,82	20,7°	350,0	1101,7	362 200
jižní	299,93	20,6°	170,2	1913,0	573 761



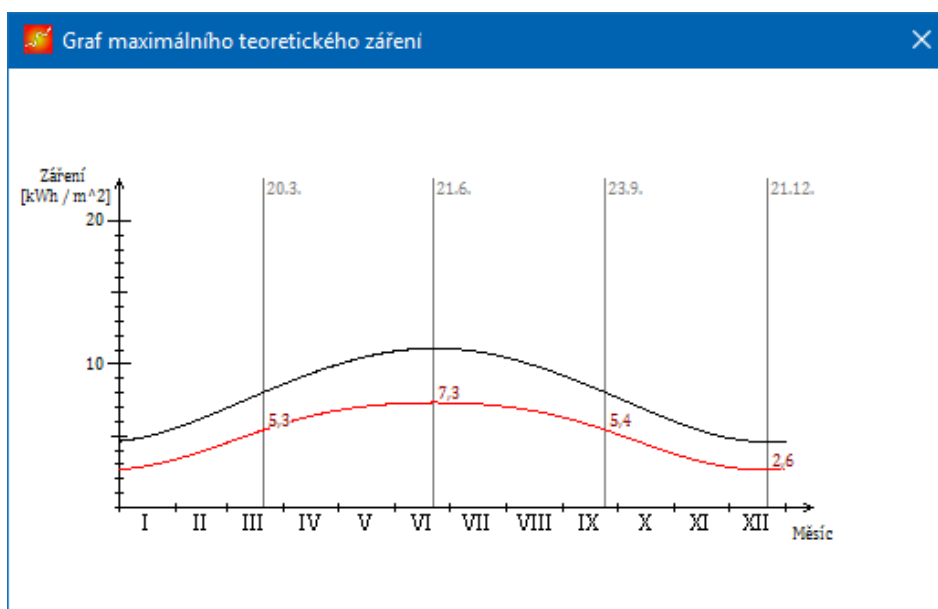
Obrázek 25: Panel solární kalkulačky



Obrázek 26: Skenovaný dům na Google mapách



Obrázek 27: Graf závislosti záření na dnu v roce vykreslené programem CMB
Španělsko, severní střecha



Obrázek 28: Grafy závislosti záření na dnu v roce vykreslené programem CMB
Španělsko, jižní střecha

6 Závěr

Cílem této práce bylo nastudovat problematiku detekce ploch v mraku bodů, popsat jednotlivé metody a provést jejich srovnání. Dále bylo za úkol vytvořit program, který by použitím jedné vybrané metody byl schopen detekovat zadané plochy.

Vzhledem k rozsáhlosti tématu se práce vymezuje pouze na detekci rovin. Nejdříve byla přiblížena metoda nejmenších čtverců, která je často využívána ostatními metodami. Dále byla popsána metoda RANSAC a některé její možné modifikace a Houghova transformace. Obě tyto metody se běžně používají k detekci ploch. Na závěr byla zmíněna i metoda rovnoběžných souřadnic, která je v současnosti používána pouze okrajově. Na závěr teoretické části byly porovnány možnosti jednotlivých metod a bylo zmíněno, že bývá vhodné kombinovat metody navzájem a přejímat tak jejich výhody.

Výstup programu byl porovnán s výstupem open source softwaru CloudCompare, neboť většina komerčních softwarů nezveřejňuje programovou dokumentaci. CloudCompare dosahoval stejných výsledků pro testované mračno jako vlastní program.

Vlastní přínos této práce je především ve vytvoření programu, který je za pomoci metody RANSAC schopen detekovat roviny v zadaném mračnu. Pokud zadaný mrak bodů představuje dům, je program navíc schopen rozeznat střechy. Nad rámec zadání byl do programu vložen výpočetní algoritmus, jehož výsledek lze použít jako pomocný ukazatel, zda se na střechy vyplatí instalovat solární panely. Je tak nastíněno i možné praktické využití získaných poznatků.

Na práci je možné navázat především rozšířením na jiné druhy ploch, nebo nalezením a následným zpracováním některých dalších existujících metod vhodných ke clusteringu mraku bodů.

Reference

- [1] KARÁSEK, Jiří a Ladislav SKULA. *Lineární algebra: teoretická část*. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3100-8.
- [2] ČERMÁK, Libor a Rudolf HLAVIČKA. *Numerické metody*. Druhé vydání. Brno: Akademické nakladatelství CERM, 2008. ISBN 978-80-214-3752-4.
- [3] MARTIŠEK, Dalibor. *Matematické principy grafických systémů*. Brno: Littera, 2002. ISBN 80-857-6319-2.
- [4] The Convex Hull of Planar Point Set. *Geometry Algorithms* [online]. 2012 [cit. 2017-03-14]. Dostupné z: http://geomalgorithms.com/a10-_hull-1.html
- [5] Shoelace formula. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-03-14]. Dostupné z: https://en.wikipedia.org/wiki/Shoelace_formula
- [6] FISCHLER, Martin A. a Robert C. BOLLERS. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* [online]. 1981, **24**(6), 381-395 [cit. 2017-03-18]. DOI: 10.1145/358669.358692. Dostupné z: <http://portal.acm.org/citation.cfm?doid=358669.358692>
- [7] Random Sample Consensus. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-03-24]. Dostupné z: https://en.wikipedia.org/wiki/Random_sample_consensus
- [8] POSPÍŠIL, Lukáš. *Skládání panoramat z fotografií*. Brno, 2011. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Pavel Svoboda.
- [9] ŘÍHA, Kamil a Petr HUJKA. Epipolární geometrie. *Elektrorevue* [online]. 2005, **2005**(17) [cit. 2017-03-24]. Dostupné z: <http://www.elektrorevue.cz/clanky/05017/index.html>
- [10] KATOCH, Rohan. Project 3 : Camera Calibration and Fundamental Matrix Estimation with RANSAC. In: *Georgia College of Tech Computing* [online]. [cit. 2017-04-02]. Dostupné z: <http://www.cc.gatech.edu/hays/compvision/results/proj3/html/pbhutada6/>
- [11] *Point Cloud Library* [online]. 2014 [cit. 2017-03-24]. Dostupné z: <http://pointclouds.org/>
- [12] TORR, P. H. S. a A. ZISSERMAN. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding* [online]. 2000, **78**, 138-156 [cit. 2017-03-23]. DOI: 10.1006/cviu.1999.0832. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S1077314299908329?via%3Dihub>
- [13] CHUM, O. a Jiří MATAS. Matching with PROSAC - Progressive Sample Consensus. *Computer Society Conference on Computer Vision and Pattern Recognition* [online]. 2005 [cit. 2017-03-24]. DOI: 10.1109/CVPR.2005.221. Dostupné

- z: https://www.researchgate.net/publication/4156175_Matching_with_PROSAC_-_progressive_sample_consensus
- [14] CHUM, O. a Jiří MATAS. Optimal Randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online]. 2008, **30**(8), 1472-1482 [cit. 2017-04-06]. DOI: 10.1109/TPAMI.2007.70787. ISSN 0162-8828. Dostupné z: <http://ieeexplore.ieee.org/document/4359381/>
 - [15] DUDA, Richard O. a Peter E. HART. *Pattern classification and scene analysis*. New York: Wiley, 1973. ISBN 04-712-2361-1.
 - [16] BORRMANN, Dorit, Jan ELSEBERG, Kai LINGEMANN a Andreas NÜCHTER. The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design. *3D Research* [online]. 2011, **2**(2), - [cit. 2017-04-18]. DOI: 10.1007/3DRes.02(2011)3. ISSN 2092-6731. Dostupné z: [http://link.springer.com/10.1007/3DRes.02\(2011\)3](http://link.springer.com/10.1007/3DRes.02(2011)3)
 - [17] Hough transform. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-18]. Dostupné z: https://en.wikipedia.org/wiki/Hough_transform
 - [18] Parallel Coordinates – How it happened. *Home of Parallel Coordinates* [online]. [cit. 2017-04-18]. Dostupné z: <http://www.math.tau.ac.il/~aiisreal/>
 - [19] ARTERO, Almir Olivette, Maria Cristina FERREIRA DE OLIVEIRA a Haim LEVKOWITZ. Uncovering Clusters in Crowded Parallel Coordinates Visualizations. *Proceeding INFOVIS '04: Proceedings of the IEEE Symposium on Information Visualization*. 2004, , 81-88. DOI: 10.1109/INFOVIS.2004.68.
 - [20] Parallel coordinates. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2017-04-06]. Dostupné z: https://en.wikipedia.org/wiki/Parallel_coordinates
 - [21] KOLOMAZNÍK, Jan. *Aplikace moderních metod pro identifikaci obrazových dat*. Brno, 2014. Disertační práce. Mendelova univerzita v Brně, Provozně ekonomická fakulta.
 - [22] AwakeKnowing/GnuplotCSharp. *GitHub* [online]. [cit. 2017-05-05]. Dostupné z: <https://github.com/AwokeKnowing/GnuplotCSharp>
 - [23] Gipuzkoa. *Infraestructura de Datos Espaciales de Gipuzkoa* [online]. [cit. 2017-05-05]. Dostupné z: http://b5m.gipuzkoa.net/url5000/en/G_22485/PUBLI&consulta=HAZLIDAR
 - [24] *Photovoltaic Education* [online]. 2008 [cit. 2017-02-20]. Dostupné z: www.pveducation.org
 - [25] Základní pojmy. *Nebeská mechanika* [online]. 2001 [cit. 2017-03-05]. Dostupné z: <http://nebmech.astronomy.cz/POJMY/pojmy.htm>
 - [26] Module sample_consensus. In: *Point Cloud Library* [online]. [cit. 2017-05-06]. Dostupné z: http://docs.pointclouds.org/trunk/group__sample__consensus.html

7 Seznam příloh

Seznam příloh na CD

CD obsahuje tři adresáře se soubory:

- 1_program
 - CMB.exe – spustitelná verze programu
 - data.txt – vzorový vstupní soubor s body
 - složka data – soubory programu
- 2_kód
 - složka CMB – zdrojové kódy programu, které lze prohlížet například pomocí programu Visual Studio 2015
- 3_gnuplot
 - gnuplot.zip – portable verze programu gnuplot potřebná k 3D vizualizaci
 - gnuplot.txt – instrukce k úspěšnému použití gnuplotu v rámci programu